RTPA™

Real-Time Program Audit™

Ability to Understand Complex Applications, Fix Bugs, and Test New Code Means Less Downtime and Reduced Costs for Clothing Manufacturer

When it comes to understanding and maintaining complex production systems, such as the one at this wholesale clothing manufacturer, the stakes are high. Downtime can cost thousands of dollars for each hour that a system is off-line. Similarly, mistakes can cause customer charge-backs or other penalties, and even a single bug may cost as much as a programmer's annual salary. Not surprisingly, programmers are under pressure to find and fix problems quickly.

Understanding and enhancing millions of lines of complex vendor application software and solving bugs in production systems can be difficult for several reasons. Failures can be intermittent and therefore not easily replicated in a test environment; reviewing the source code's logic may not reveal where or why the bug occurs; the bug may be the result of a complex interaction between homegrown code and the vendor-supplied software; or the failure may occur only in response to real-world data sets that are difficult or impossible to replicate in a test setting.

This is the working environment of Jerry Hollo, an experienced Senior Programmer/Analyst, for a large clothing wholesaler. Much of his work is with Manhattan Associates Warehouse management System (PkMS), a very large complex supply chain execution platform with large RPG programs and multiple program calls to numerous sub-programs.

Jerry was often faced with the challenge of testing new code that had been added to in-house programs that perform functions specific for his company. These new programs ride on top of the production program environment. Because of the complex interaction between the in-house programs and the vendor-supplied programs, when bugs occur, it is often difficult to determine what is causing the problem. The error could be a result of a side-effect of a new business rule, where data changed in the in-house program inadvertently affects the function of the vendor-supplied program. Regardless of the cause, these problems must be handled quickly. If a developer spends too much time finding bugs, it could adversely impact on deadlines for other corporate projects.

Jerry used RTPA for IBM i RPG to address his challenges. The debugging process took much longer before he had RTPA. "Sometimes I used before and after data analysis. I would also insert calls to a temporary program that would print out limited pre-selected data. I would use debug if the program was interactive, although this involved the additional step of submitting jobs to batch processing."

AT-A-GLANCE

Industry:
Manufacturing

Type:
Apparel

Problems solved:

- Find and fix bugs in complex code
- Test and develop new programs, faster
- Understand and enhance large complex vendor application software
- Minimize system downtime for better operational and financial efficiency

Problems solved:

Real-Time Program Audit for IBM i RPG

Results:

- Reduced system downtime resulted in cost savings
- Increased productivity and time saved for IT staff in debugging, testing, and analysing
- Higher levels of customer satisfaction with system

RTPA™

Real-Time Program Audit™

"With RTPA, I am able to see everything that happens in all of the programs. I can immediately see everywhere that a piece of data is used, which business rules are put into operations, their results and all of their intermediate values."

Debugging PkMS often involves following the program's execution flow through numerous levels based on the values of various parameters. One program may or may not get called depending on the value of one of the parameters. One bug that Jerry encountered was the result of placing code in the wrong program in a stack of eleven programs. Jerry credited RTPA with dramatically improving his productivity.

Jerry uses RTPA to audit-enable the programs in the production system to resolve critical production problems. By audit-enabling all of the programs that are being executed, he can find and solve problems faster than other options such as a traditional debugger. The following specific data analytics strategies enabled by RTPA program-auditing functions enabled Jerry to quickly accomplish his goals:

- Auditing a unique variable, such as a customer number, and searching the audit file. This saved time by allowing him to skip over normal cases.
- Auditing the time of execution along with the source statements. Doing this allowed Jerry to match the audit result with the time of failure.
- Setting up a test system and engaging the audit until a bug occurred. This 'hands off' approach allowed him to get other work done while waiting for errors.

> *"Everything worked fine under one condition, but failed to work under another. Using RTPA helped me to understand that I needed to place my code in a different program so that it would execute under both circumstances."*
>
> *"I was able to save about 40 hours of debugging/testing/analysis time. With RTPA I was able to solve most problems within 1-2 hours." Without RTPA, Jerry said it could often take from one day to two weeks, depending on the complexity of the programs involved.*

Using RTPA's multi-faceted analytical capabilities, Jerry was able to accelerate the debugging process and test new code faster and more efficiently, including interactions between in-house and vendor-supplied programs. For the company, this resulted in significantly reduced costs spent for programmers' time, and the ability to fix more problems, more quickly, resulting in an overall better system. The end result was a system with fewer failures and shorter outages. Less downtime and fewer glitches means higher levels of satisfaction for both the company and the customers it serves.

FIND OUT MORE

Visit www.realtimeprogramaudit.com or email info@harkinsaudit.com
for more information about how to get started with RTPA now.

IBM® Registered Business Partner™

Record and understand the real-time flow of your code.
See what executed. Know what happened.